Osmangazi Tıp Dergisi
Osmangazi Journal of Medicine 2020; 42 (1):103-109

Research Article / Araştırma Makalesi

A Study on Missing Data Problem in Random Forest

Rasgele Orman Yönteminde Eksik Veri Probleminin İncelenmesi

Hulya Ozen, Cengiz Bal

Department of Biostatistics, Faculty of Medicine, Eskisehir Osmangazi University, Eskisehir, Turkey

Abstract: Random Forest is an ensemble method that combines many trees constructed from bootstrap samples of the original data. Random Forest is used for both classification and regression and provides many advantages such as having a high accuracy, calculating a generalization error, determining the important variables and outliers, performing supervised and unsupervised learning and imputing missing values with an algorithm based on proximity matrix. In this study, we aimed to compare the proximity based imputation method of Random Forest with k nearest neighbor imputation prior to fitting. Therefore, simulation studies were performed for a classification problem under various scenarios including different percentage of missing values, number of neighbors and correlation structures between predictor variables. The results showed that for highly correlated structures proximity matrix based imputation method should be used meanwhile k nearest neighbor imputation method should be preferred for low and medium correlated structures.
Keywords: knn imputation method, missing value, proximity matrix, random forest

Özet: Rasgele Orman, orijinal verilerin bootstrap örneklerinden oluşturulmuş pek çok karar ağacını bir araya getiren bir topluluk yöntemidir. Rasgele Orman, hem sınıflandırma hem de regresyon için kullanılır ve yüksek doğruluk oranı elde etme, genelleme hatası hesaplama, önemli değişkenleri ve aykırı değerleri belirleme, danışmanlı ve danışmansız öğrenmeyi gerçekleştirme ve yakınlık matrisine dayalı bir algoritma ile eksik gözlemlere değer atama gibi birçok avantaj sağlar. Bu çalışmada, Rasgele Orman'ın yakınlık matrisi temelli atama yöntemini, model kurulumundan önce kullanılan en yakın komşu ile değer atama yöntemiyle karşılaştırmayı amaçladık. Bu nedenle, farklı eksik değer yüzdeleri, komşuluk sayısı ve tahminci değişkenler arasındaki korelasyon yapıları dahil olmak üzere çeşitli senaryolar altında bir sınıflandırma problemi için simülasyon çalışması yapılmıştır. Sonuçlar, yüksek korelasyonlu yapılar için yakınlık matrisi tabanlı atama yönteminin kullanılması gerektiğini, orta ve düşük korelasyonlu yapılar için ise en yakın komşu ile değer atama yönteminin tercih edilmesi gerektiğini göstermektedir.
Anahtar Kelimeler: knn atama yöntemi, eksik veri, yakınlık matrisi, rasgele orman

ORCID ID of the authors: H.Ö 0000-0003-4144-3732; C.B. 0000-0002-1553-2902

Received 13.12.2018 ***Accepted*** 01.03.2019 ***Online published*** 04.03.2019

Correspondence: **Hülya ÖZEN**- Department of Biostatistics, Faculty of Medicine, Eskisehir Osmangazi University, Eskisehir, Turkey
e-mail: hulya_ozen@yahoo.com

Cite this article as:
Ozen H, Bal C. A Study on Missing Data Problem in Random Forest, Osmangazi Journal of Medicine, 2020;42(1):103-109
Doi: *10.20515/otd.496524*

1. Introduction

Random Forest (RF) is an ensemble learning method that combines the results of decision trees generated by selecting samples from the same data set by bootstrap method and can be used for both classification and regression purposes (1, 2). RF is commonly used in areas such as ecology, genetics, bioinformatics where the high-dimensional data takes place (3-6). RF can perform supervised or unsupervised learning. RF uses *m* variables, where m is less than the number of all predictor variables *p*, while splitting the nodes to create different trees and overcome the overfitting problem. In a classification algorithm *m* is equal to $\sqrt{p}$ and it is p/3 for a regression algorithm. Also, RF can give a generalization error for all trees in the forest. It provides not only an intuitive measure of variable importance, but also a proximity matrix that gives the distances between the observations. Moreover it can handle missing value problems with an algorithm based on proximities (1, 2). Among the many other methods in data mining, RF provides superior imputation results (7). Missing data problems in a RF algorithm can also be solved by imputing the data with some methods before constructing the trees. Single imputation methods by mean, median, hot deck, cold deck or linear regression are no longer used since they tend to underestimate the variance (8, 9). Therefore, new approaches are developed. K Nearest Neighbor (KNN) imputation is one of the most preferred methods in literature. It is based on the distance between observations and commonly used for high dimensional data such as microarrays (10, 11). Although both KNN and proximity matrix based missing value imputation approaches are popular, it has not been studied if one of the methods is fairly superior to the other.

The aim of this study was to compare the proximity matrix based imputation method with KNN imputation method prior to constructing the forests in a classification problem. In concordance with the purpose of this study, comparisons were made through the simulation results. In Section 2, we mentioned the methodology and the construction of simulation algorithm. Simulation results were presented in tables in Section 3. Finally, discussion with other studies and conclusions were detailed in Section 4. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

2. Methodology

2.1 Generating Full and Missing Data Sets

A categorical response variable was created since the purpose of this study was to compare methods in a classification problem. All predictor variables were determined to be continuous variables. Each dataset were decided to consist of one categorical response and 30 predictor variables for each iteration.

Predictor variables were generated by dividing the data set into two different parts. In the first part, five of the predictor variables were generated from the multivariate normal distribution with mean 0 and variance-covariance matrix $\boldsymbol{\Sigma}_i$, which were shown below:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 1 \end{bmatrix}$$

In order to create low, medium and highly correlated structures among the predictor variables, off-diagonal elements of variance-covariance matrices were chosen as 0.1, 0.5 and 0.9, respectively (12). In the second part, the rest of the predictor variables were generated from multivariate normal distribution with mean 0 and identity matrix as variance-covariance matrix, so that they were determined to be uncorrelated. By doing so, it was provided to split the nodes by the first part of the predictor variables.

Let y be the response variable which is binary with the values 0 and 1. The response variable was obtained in the following steps: First; a binary logistic regression model was used to have a $\pi(X)$ vector that was shown in (1):

$$P(y=1|X)=\pi(X)=\frac{exp(X^T\beta)}{1+exp(X^T\beta)}(1)$$

Here X was a vector which contained the first part of the predictor variables and the effects of the X's on $\pi(X)$ was set to be equal $\beta^T = [1,2,3,4,5]$, while β_0 assumed to be 0. After obtaining, $\pi(X)$, the probability values, they were put into the inverse cumulative distribution function of Bernoulli. In doing so, class labels were obtained as 0 and 1.With merging the response and predictor variables, the simulated data set took the final form.

After we had obtained the full dataset, missing values were created on the two of predictor variables from the first part to meet missing values during construction of the forests. In order to have a missing completely at random mechanism, we used random sampling method without replacement. Missing values were created on both variables separately with the same percentage. Tronskaya et. al and Rieger et. al (11, 12) worked with the upper bound of missing percentage as %20 for a predictor variable. Rieger et. al also led at the maximum 50% of data set contain missing values. When these studies were considered, we determined to study the percentages of missing values for both predictor variables as 5%, 10%, 15%, 20% and 25% in the simulations.

2.2 The Missing Value Imputation Methods

Two missing value imputation approaches were compared in this study. The first method was the missing value imputation algorithm of RF which was based on proximity matrix. RF calculates a (nxn) proximity matrix to evaluate the similarity of observations. Off- diagonal elements of the matrix gives the similarity of two different observations. Based on these proximity values, RF carries out an iterative process for imputation by following these steps: first an initial forest is built after using median imputation and then proximities are calculated. New imputed values are calculated by a proximity based weighted mean. With this updated data set, a new forest is built and by doing so new proximities and imputed values are obtained. It is found that after performing 5 or 6 iterations, sufficient results can be seen (13). In this study, number of iteration was determined to be 5. While this proximity based imputation method is applied during building a forest, the second approach in the study, KNN imputation method, was applied to data set before fitting the RF. In KNN imputation method, first the neighbors are determined by calculating the distance measures between observations. These measures are obtained through Minkowski, Manhattan or Euclidean functions. Because of being the most popular one amongst the others, Euclidean distance function was used in this study. Later, imputations are done based on weighted mean values of k nearest neighbors. The weights are inversely proportional to the distance measures. Not only different distance functions, but also different algorithms of KNN can be seen in literature. Some of them do not permit the neighbor values to contain missing values (14-16). But this might cause the method to give less efficient results. In this study, the KNN algorithm in R package "impute" was used. This method presents more notable results than the ones mentioned above (10). The values of k, the number of nearest neighbors, were determined as k=5, 10, 15 and 20 for the simulation studies (11).

2.3 Simulation Design

In this study (100000/n) Monte Carlo simulation technique was performed with R package program. Sample sizes were determined as n=100, 200, 500 and 1000 and number of simulations were taken as s=1000, 500, 200 and 100, respectively. In the simulation studies, all possible combinations of sample sizes, correlation structures, percentages of missing values and numbers of nearest neighbors were evaluated with an algorithm.

The algorithm was built through the following steps: First a full data set was generated. Then missing values was created on data set for various percentages. Imputation methods mentioned above were applied on the same data sets with missing values separately. In doing so, five different imputed data sets were obtained besides the full data set. Classification tables were obtained, after all of the data sets had been put into the same RF algorithm separately. We used 500 trees for each RF algorithm and the number of nodes to sample at each split was decided to be equal to $\sqrt{p}$ which is equal to 5 in this study (1-3). In order to have the same RF algorithms for all imputed data sets, the same seed numbers were used. since all the simulations were based on classification problems, true classification rates (TCR) were calculated through the Table 1 by the formula (a+d)/n.

Table 1. Classification table of true and predictive values

		Predictive Classes		Total
		0	1	
True Classes	0	a	b	a+b
	1	c	d	c+d
Total		a+c	b+d	n

Imputation methods were compared with each other through TCR values. The method giving the closest result to the TCR value of full data set was chosen as the best among the others.

3. Results

Simulation results were given in Table 2-4. TCR results for low correlated simulated data were shown in Table 2. All the methods presented close results but, KNN was better in the case of k=15 and k=20.

In Table 3, results for medium correlated simulated data were given. Similar results were observed in Table 3, and as in Table 2 for the value of k=15 and k=20, KNN gave better results among the others. Results of highly correlated simulated data were shown in Table 4. Unlike the other results, proximity matrix showed better performance where the sample size was greater than 100. Considering all the results in Table 2-4, it was obvious that all imputation methods showed close results not only to each other, but also to the TCR of full data sets.

Table 2.TCR results of low correlated simulated data after imputation methods applied

		Full Data Set	Proximity Matrix	KNN Imputation			
				K=5	K=10	K=15	K=20
n=100	5%	**0.82738**	0.81771	0.81757	0.81811	**0.81843**	0.81840
	10%		0.81096	0.81082	0.81043	0.81118	**0.81183**
	15%		0.80173	0.80259	0.80411	**0.80446**	0.80444
	20%		0.79225	0.79330	0.79483	0.79399	**0.79488**
	25%		0.78271	0.78492	0.78508	0.78687	**0.78700**
n=200	5%	**0.86116**	0.85314	0.85342	**0.85376**	0.85363	0.85262
	10%		0.84550	0.84454	0.84643	**0.84731**	0.84629
	15%		0.83576	0.83790	0.83886	0.83898	**0.83917**
	20%		0.82754	0.82842	0.83029	**0.83251**	0.83148
	25%		0.82082	0.82314	0.82478	**0.82586**	0.82584
n=500	5%	**0.88748**	0.87778	0.87935	0.88008	**0.88020**	0.88019
	10%		0.87133	0.87190	0.87251	**0.87286**	0.87270
	15%		0.86396	0.86452	0.86513	**0.86579**	0.86462
	20%		0.85615	0.85698	0.85744	0.85858	**0.85956**
	25%		0.84774	0.85063	0.85105	0.85068	**0.85180**
n=1000	5%	**0.90039**	0.89277	0.89184	0.89285	**0.89361**	0.89327
	10%		0.88555	0.88625	0.88569	**0.88705**	0.88676
	15%		0.87845	0.87919	0.87961	**0.87973**	0.87918
	20%		0.86838	0.87082	**0.87148**	0.87059	0.87045
	25%		0.86228	0.86429	0.86612	**0.86615**	0.86544

Table 3. TCR results of medium correlated simulated data after imputation methods applied

		Full Data Set	Proximity Matrix	KNN Imputation			
				K=5	K=10	K=15	K=20
n=100	5%	**0.90723**	0.90273	0.90276	0.90262	**0.90332**	0.90261
	10%		0.89773	0.89730	0.89855	0.89908	**0.89947**
	15%		0.89395	0.89470	0.89601	**0.89682**	0.89622
	20%		0.88936	0.88902	0.89064	0.89159	**0.89175**
	25%		0.88325	0.88443	0.88576	**0.88780**	0.88706
n=200	5%	**0.92070**	0.91632	0.91582	0.91683	**0.91687**	0.91600
	10%		0.91221	0.91172	**0.91264**	0.91178	0.91237
	15%		0.90914	0.90864	0.90924	**0.91000**	0.90993
	20%		0.90376	0.90224	0.90364	0.90489	**0.90507**
	25%		0.89934	0.89762	0.90042	0.90039	**0.90100**
n=500	5%	**0.93370**	0.92915	0.92856	0.92927	**0.92979**	0.92914
	10%		0.92384	0.92389	0.92536	0.92557	**0.92581**
	15%		0.92080	0.91942	0.92131	**0.92175**	0.92166
	20%		0.91538	0.91365	0.91627	**0.91705**	0.91598
	25%		0.91327	0.91050	0.91282	0.91298	**0.91442**
n=1000	5%	**0.93804**	0.93390	0.93324	**0.93436**	0.93353	0.93376
	10%		0.92994	0.93014	0.92994	0.92987	**0.93027**
	15%		0.92563	0.92468	0.92590	0.92617	**0.92684**
	20%		**0.92229**	0.91939	0.92220	0.92146	0.92220
	25%		0.91870	0.91539	0.91726	0.91824	**0.91878**

Table 4. TCR results of highly correlated simulated data after imputation methods applied

		Full Data Set	Proximity Matrix	KNN Imputation			
				K=5	K=10	K=15	K=20
n=100	5%	**0.94761**	0.94543	0.94599	0.94615	**0.94641**	0.94616
	10%		0.94407	0.94367	0.94395	0.94445	**0.94481**
	15%		0.94329	0.94229	0.94300	0.94319	**0.94375**
	20%		0.94205	0.94134	0.94110	0.94189	**0.94284**
	25%		0.94035	0.93942	0.94055	**0.94132**	0.94088
n=200	5%	**0.95111**	0.94958	0.94903	0.94975	0.94962	**0.94982**
	10%		**0.94820**	0.94761	0.94764	0.94784	0.94800
	15%		**0.94771**	0.94608	0.94602	0.94683	0.94716
	20%		**0.94624**	0.94453	0.94484	0.94559	0.94563
	25%		**0.94543**	0.94319	0.94370	0.94389	0.94469
n=500	5%	**0.95525**	0.95392	0.95292	**0.95394**	0.95393	0.95389
	10%		**0.95394**	0.95210	0.95230	0.95257	0.95366
	15%		**0.95153**	0.95012	0.95130	0.95097	0.95127
	20%		**0.95112**	0.94893	0.94905	0.94972	0.95013
	25%		**0.94995**	0.94702	0.94748	0.94874	0.94912
n=1000	5%	**0.95750**	**0.95657**	0.95559	0.95595	0.95653	0.95620
	10%		**0.95473**	0.95374	0.95397	0.95473	0.95463
	15%		**0.95422**	0.95161	0.95239	0.95307	0.95313
	20%		**0.95282**	0.95048	0.95116	0.95175	0.95195
	25%		**0.95195**	0.94878	0.94957	0.94996	0.95037

Conflict of Interest: *The authors declare that they have no conflict of interest.*

The increase in both the sample size and correlation between variables increased the TCR values. On the contrary, the increase in percentage of missing value affected the TCR values in the opposite direction. KNN imputation method presented a good performance for the values where k was equal to at least 10. Highly correlated structure made proximity matrix give better results. It was clear that, the lowest TCR results were obtained by KNN method where k value was equal to 5.

4. Discussion and Conclusion

In this paper, we studied the missing value problem for the RF algorithm. The data sets with different sample sizes and correlation structures were generated, then missing values were created randomly on these data sets. Later on, imputation with proximity matrix and KNN method for various k values were compared with each other in different scenarios.

Scheel, Aldrin (10) used KNN imputation for microarray data and compared it with a method they proposed. Troyanskaya, Cantor (11) also studied the missing value problem in microarray data sets and they proposed not to use small k values for KNN imputation method. Acuna and Rodriguez (14) also suggested to avoid from small k values to prevent inefficient imputations based on the dominant observations. Rieger, Hothorn (12) compared the KNN imputation method, where k was equal to 10, with surrogate variables of the Conditional Inference Trees (CIF) which has an algorithm close to RF. They showed that both KNN imputation and surrogate variables gave similar results and had no superiority on each other.

In our study, TCR results were obtained after the imputation with both proximity matrix and KNN method. As we increased the sample size and correlation among the important variables, we observed an increase in TCR values for both full and imputed data sets. Also the difference between TCR values of full and imputed data sets decreased. However, the increase in percentage of missing values on the predictor variables caused the results to decrease. As in Rieger, Hothorn (12), methods presented similar performance in the simulations, but they were also observed to be superior to each other. Especially, correlation among the predictor variables created an important effect. For low and medium correlated simulated data sets, KNN imputation method resulted in better performance. The value of k should be at least equal to 10, since the results in case of k=5 was the smallest among the others. The performance of proximity matrix was better when the correlation structure was high in the data sets. In practice, obtaining the correlation between important variables can give an idea about the correlation structure of a data set. Important variables can be found, after calculating the variable importance measures through a RF algorithm.

In conclusion, to have a well-imputed data set for a RF algorithm, correlation structure must be taken into consideration. KNN should be preferred where the data set has low or medium correlated structure meanwhile for highly correlated structures proximity matrix should be used.

REFERENCES

1. Breiman L. Random forests. *Mach Learn.* 2001;45:5-32.
2. Cutler A, Cutler D, Stevens J, Zhang C, Ma Y. Ensemble Machine Learning: Methods and Applications. Springer Science+ Business Media, LLC; 2012.
3. Qi Y. Random forest for bioinformatics. Ensemble machine learning: Springer; 2012;p. 307-23.
4. Moorthy K, Mohamad MS, Deris S, editors. Multiclass Prediction for Cancer Microarray Data Using Various Variables Range Selection Based on Random Forest. Pacific-Asia Conference on Knowledge Discovery and Data Mining; 2013: Springer.
5. Wu X, Wu Z, Li K, editors. Classification and identification of differential gene expression for microarray data: improvement of the random forest method. 2008 2nd International Conference on Bioinformatics and Biomedical Engineering; 2008: IEEE.
6. Cutler DR, Edwards TC, Beard KH, Cutler A, Hess KT, Gibson J, et al. Random forests for classification in ecology. *Ecology*. 2007;88:2783-92.
7. Pantanowitz A, Marwala T. Evaluating the Impact of Missing Data Imputation through the use of the Random Forest Algorithm. arXiv preprint arXiv:08122412. 2008.
8. Soley-Bori M. Dealing with missing data: Key assumptions and methods for applied analysis. Boston University. 2013.
9. Takahashi M, Ito T. Multiple Imputation of Turnover in EDINET Data: Toward the Improvement of Imputation for the Economic Census. Work Session on Statistical Data Editing, UNECE. 2012:24-6.
10. Scheel I, Aldrin M, Glad IK, Sorum R, Lyng H, Frigessi A. The influence of missing value

imputation on detection of differentially expressed genes from microarray data. *Bioinformatics*. 2005;21:4272-9.

11. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*. 2001;17:520-5.
12. Rieger A, Hothorn T, Strobl C. Random forests with missing values in the covariates. 2010.
13. Breiman L, Cutler A. RFtools—for predicting and understanding data. Berkeley, CA, USA, Tech Rep. 2004.
14. Acuna E, Rodriguez C. The treatment of missing values and its effect on classifier accuracy. Classification, clustering, and data mining applications: Springer; 2004; p. 639-47.
15. Lee M-LT. Analysis of microarray gene expression data: Springer Science & Business Media; 2007.
16. Ouyang M, Welsh WJ, Georgopoulos P. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics.* 2004;20:917-23.